



OpenShift Container Platform 4.3

IBM Z へのインストール

OpenShift Container Platform 4.3 IBM Z クラスターのインストール

OpenShift Container Platform 4.3 IBM Z へのインストール

OpenShift Container Platform 4.3 IBM Z クラスターのインストール

法律上の通知

Copyright © 2020 Red Hat, Inc.

The text of and illustrations in this document are licensed by Red Hat under a Creative Commons Attribution–Share Alike 3.0 Unported license ("CC-BY-SA"). An explanation of CC-BY-SA is available at

<http://creativecommons.org/licenses/by-sa/3.0/>

. In accordance with CC-BY-SA, if you distribute this document or an adaptation of it, you must provide the URL for the original version.

Red Hat, as the licensor of this document, waives the right to enforce, and agrees not to assert, Section 4d of CC-BY-SA to the fullest extent permitted by applicable law.

Red Hat, Red Hat Enterprise Linux, the Shadowman logo, the Red Hat logo, JBoss, OpenShift, Fedora, the Infinity logo, and RHCE are trademarks of Red Hat, Inc., registered in the United States and other countries.

Linux[®] is the registered trademark of Linus Torvalds in the United States and other countries.

Java[®] is a registered trademark of Oracle and/or its affiliates.

XFS[®] is a trademark of Silicon Graphics International Corp. or its subsidiaries in the United States and/or other countries.

MySQL[®] is a registered trademark of MySQL AB in the United States, the European Union and other countries.

Node.js[®] is an official trademark of Joyent. Red Hat is not formally related to or endorsed by the official Joyent Node.js open source or commercial project.

The OpenStack[®] Word Mark and OpenStack logo are either registered trademarks/service marks or trademarks/service marks of the OpenStack Foundation, in the United States and other countries and are used with the OpenStack Foundation's permission. We are not affiliated with, endorsed or sponsored by the OpenStack Foundation, or the OpenStack community.

All other trademarks are the property of their respective owners.

概要

本書では、IBM Z に OpenShift Container Platform 4.3 クラスターをインストールする方法について説明します。

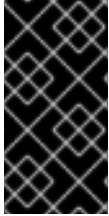
目次

第1章 IBM Zへのインストール	3
1.1. クラスターのIBM Zへのインストール	3

第1章 IBM Z へのインストール

1.1. クラスターの IBM Z へのインストール

OpenShift Container Platform version 4.3 では、プロビジョニングする IBM Z インフラストラクチャーにクラスターをインストールできます。



重要

ベアメタルプラットフォーム以外の場合には、追加の考慮点を検討する必要があります。OpenShift Container Platform クラスターをインストールする前に、「[guidelines for deploying OpenShift Container Platform on non-tested platforms](#)」にある情報を確認してください。

前提条件

- クラスターの [NFS を使用する永続ストレージ](#) をプロビジョニングします。プライベートイメージレジストリーをデプロイするには、ストレージで ReadWriteMany アクセスモードを指定する必要があります。
- [OpenShift Container Platform のインストールおよび更新](#) プロセスについての詳細を確認します。
- ファイアウォールを使用する場合、クラスターがアクセスを必要とする [サイトを許可するようにファイアウォールを設定](#) する必要があります。



注記

プロキシを設定する場合は、このサイト一覧も確認してください。

1.1.1. OpenShift Container Platform のインターネットアクセスおよび Telemetry アクセス

OpenShift Container Platform 4.3 では、クラスターをインストールし、有効なサブスクリプションでクラスターを使用するにはインターネットアクセスが必要になります。クラスターの健全性および正常に実行された更新についてのメトリクスを提供するためにデフォルトで実行される Telemetry サービスにもインターネットアクセスが必要です。クラスターがインターネットに接続されている場合、Telemetry は自動的に実行され、クラスターは [Red Hat OpenShift Cluster Manager](#) に登録されます。ここで、エンタイトルメントをクラスターに割り当てることができます。

インターネットへのアクセスは以下を実行するために必要です。

- [Red Hat OpenShift Cluster Manager](#) ページにアクセスし、インストールプログラムをダウンロードし、サブスクリプション管理およびエンタイトルメントを実行します。クラスターにインターネットアクセスがあり、Telemetry を無効にしない場合、そのサービスは有効なサブスクリプションでクラスターを自動的に使用します。Telemetry サービスが有効なサブスクリプションでクラスターを使用しない場合、[Cluster registration](#) ページでこれを手動で可能にする必要があります。
- クラスターのインストールに必要なパッケージを取得するために [Quay.io](#) にアクセスします。
- クラスターの更新を実行するために必要なパッケージを取得します。



重要

クラスターでインターネットに直接アクセスできない場合、プロビジョニングするインフラストラクチャーでネットワークが制限されたインストールを実行できます。このプロセスで、必要なコンテンツをダウンロードし、これを使用してミラーレジストリーにクラスターのインストールおよびインストールプログラムの生成に必要なパッケージを設定します。インストールタイプによっては、クラスターのインストール環境でインターネットアクセスが不要となる場合があります。クラスターを更新する前に、ミラーレジストリーのコンテンツを更新します。

1.1.2. ユーザーによってプロビジョニングされるインフラストラクチャーを使用する場合のクラスターのマシン要件

ユーザーによってプロビジョニングされるインフラストラクチャーを含むクラスターの場合、必要なマシンすべてをデプロイする必要があります。

1.1.2.1. 必要なマシン

最小の OpenShift Container Platform クラスターでは以下のホストが必要です。

- 1つの一時的なブートストラップマシン
- 3つのコントロールプレーン、またはマスター、マシン
- 2つ以上のコンピュートまたはワーカーマシン



注記

クラスターでは、ブートストラップマシンが OpenShift Container Platform クラスターを3つのコントロールプレーンマシンにデプロイする必要があります。クラスターのインストール後にブートストラップマシンを削除できます。



重要

クラスターの高可用性を改善するには、複数の異なる z/VM インスタンスにコントロールプレーンマシンを分散します。これらは同じ Z または LinuxONE ハードウェアで実行することができますが、これは必須ではありません。

ブートストラップ、コントロールプレーンおよびコンピュートマシンでは、Red Hat Enterprise Linux CoreOS (RHCOS) をオペレーティングシステムとして使用する必要があります。

RHCOS は Red Hat Enterprise Linux 8 をベースとしており、そのハードウェア認定および要件が継承されることに注意してください。「[Red Hat Enterprise Linux テクノロジーの機能と制限](#)」を参照してください。

1.1.2.2. ネットワーク接続の要件

すべての Red Hat Enterprise Linux CoreOS (RHCOS) マシンでは、起動時に `inittmfs` のネットワークがマシン設定サーバーから Ignition 設定ファイルをフェッチする必要があります。マシンは静的 IP アドレスで設定されます。DHCP サーバーは必要ありません。

1.1.2.3. IBM Z ネットワーク接続の要件

IBM Zの z/VM でインストールするには、レイヤー 2 モードの単一 z/VM 仮想 NIC が必要になります。以下も必要になります。

- 直接接続された OSA または RoCE ネットワークアダプター
- z/VM vSwitch のセットアップ。推奨されるセットアップでは、OSA リンクアグリゲーションを使用します。

1.1.2.4. 最小リソース要件

それぞれのクラスターマシンは、以下の最小要件を満たしている必要があります。

マシン	オペレーティングシステム	vCPU	RAM	ストレージ
ブートストラップ	RHCOS	4	16 GB	120 GB
コントロールプレーン	RHCOS	4	16 GB	120 GB
コンピューター	RHCOS	2	8 GB	120 GB

1.1.2.5. 最小の IBM Z システム要件

IBM Z ハードウェアを使用する場合で、z13 以降を使用する場合は OpenShift Container Platform バージョン 4.3 をインストールできます。IBM LinuxONE ハードウェアを使用する場合は、これを LinuxONE z13 以降にインストールできます。

ハードウェア要件

- SMT2 をサポートする 3 IFL 搭載の 1 LPAR
- 1 OSA または RoCE ネットワークアダプター

オペレーティングシステム要件

- z/VM 7.1 の 1 インスタンス

z/VM インスタンスで以下をセットアップします。

- OpenShift Container Platform コントロールプレーンマシンの 3 ゲスト仮想マシン
- OpenShift Container Platform コンピューターマシンの 2 ゲスト仮想マシン
- 一時 OpenShift Container Platform ブートストラップマシンの 1 ゲスト仮想マシン

z/VM ゲスト仮想マシンのディスクストレージ

- FICON 接続のディスクストレージ (DASD) これらには z/VM ミニディスク、フルパックミニディスク、または専用の DASD を使用できます。利用可能な場合は、HyperPAV を使用して最適なパフォーマンスを確保します。
- FCP 接続のディスクストレージ

ストレージ/メインメモリー

- OpenShift Container Platform コントロールプレーンマシン用に 16 GB
- OpenShift Container Platform コンピュートマシン用に 8 GB
- 一時 OpenShift Container Platform ブートストラップマシン用に 16 GB

1.1.2.6. 推奨される IBM Z システム要件

ハードウェア要件

- SMT2 をサポートする 6 IFL 搭載の 3 LPAR
- 1 または 2 OSA または RoCE ネットワークアダプター、またはその両方

オペレーティングシステム要件

- 高可用性を確保する場合は z/VM 7.1 の 2 または 3 インスタンス

z/VM インスタンスで以下を設定します。

- OpenShift Container Platform コントロールプレーンマシン用に 3 ゲスト仮想マシン (z/VM インスタンスごとに 1 つ)
- OpenShift Container Platform コンピュートマシン用に 6 以上のゲスト仮想マシン (z/VM インスタンス全体に分散)
- 一時 OpenShift Container Platform ブートストラップマシンの 1 ゲスト仮想マシン

z/VM ゲスト仮想マシンのディスクストレージ

- FICON 接続のディスクストレージ (DASD) これらには z/VM ミニディスク、フルパックミニディスク、または専用の DASD を使用できます。利用可能な場合は、HyperPAV を使用して最適なパフォーマンスを確保します。
- FCP 接続のディスクストレージ

ストレージ/メインメモリー

- OpenShift Container Platform コントロールプレーンマシン用に 16 GB
- OpenShift Container Platform コンピュートマシン用に 8 GB
- 一時 OpenShift Container Platform ブートストラップマシン用に 16 GB

1.1.2.7. 証明書署名要求の管理

ユーザーがプロビジョニングするインフラストラクチャーを使用する場合、クラスターの自動マシン管理へのアクセスは制限されるため、インストール後にクラスターの証明書署名要求 (CSR) のメカニズムを提供する必要があります。**kube-controller-manager** は kubelet クライアント CSR のみを承認します。**machine-approver** は、kubelet 認証情報を使用して要求される提供証明書の有効性を保証できません。適切なマシンがこの要求を発行したかどうかを確認できないためです。kubelet 提供証明書の要求の有効性を検証し、それらを承認する方法を判別し、実装する必要があります。

1.1.3. ユーザーによってプロビジョニングされるインフラストラクチャーの作成

ユーザーによってプロビジョニングされるインフラストラクチャーを使用する OpenShift Container Platform クラスターをデプロイする前に、基礎となるインフラストラクチャーを作成する必要があります。

前提条件

- クラスターでサポートするインフラストラクチャーを作成する前に、「[OpenShift Container Platform 4.x Tested Integrations](#)」ページを参照してください。

手順

1. 静的 IP アドレスをセットアップします。
2. FTP サーバーをセットアップします。
3. 必要なロードバランサーをプロビジョニングします。
4. マシンのポートを設定します。
5. DNS を設定します。
6. ネットワーク接続を確認します。

1.1.3.1. ユーザーによってプロビジョニングされるインフラストラクチャーのネットワーク要件

すべての Red Hat Enterprise Linux CoreOS (RHCOS) マシンでは、起動時に **initramfs** のネットワークがマシン設定サーバーから Ignition 設定をフェッチする必要があります。

初回の起動時に、Ignition 設定ファイルをダウンロードできるようにネットワーク接続を確立するために、マシンには FTP サーバーが必要になります。

マシンに永続 IP アドレスおよびホスト名があることを確認します。

Kubernetes API サーバーはクラスターマシンのノード名を解決する必要があります。API サーバーおよびワーカーノードが異なるゾーンに置かれている場合、デフォルトの DNS 検索ゾーンを、API サーバーでノード名を解決できるように設定することができます。もう1つの実行可能な方法として、ノードオブジェクトとすべての DNS 要求の両方において、ホストを完全修飾ドメイン名で常に参照することができます。

マシン間のネットワーク接続を、クラスターのコンポーネントが通信できるように設定する必要があります。すべてのマシンではクラスターの他のすべてのマシンのホスト名を解決する必要があります。

表1.1 すべてのマシンに対応するすべてのマシン

プロトコル	ポート	説明
TCP	2379-2380	etcd サーバー、ピア、およびメトリクスポート
	6443	Kubernetes API
	9000-9999	ホストレベルのサービス。ポート 9100-9101 のノードエクスポーター、ポート 9099 の Cluster Version Operator が含まれます。

プロトコル	ポート	説明
	10249-10259	Kubernetes が予約するデフォルトポート
	10256	openshift-sdn
UDP	4789	VXLAN および GENEVE
	6081	VXLAN および GENEVE
	9000-9999	ポート 9100-9101 のノードエクスポーターを含む、ホストレベルのサービス。
	30000-32767	Kubernetes NodePort

ネットワークポロジリー要件

クラスター用にプロビジョニングするインフラストラクチャーは、ネットワークポロジリーの以下の要件を満たす必要があります。



重要

OpenShift Container Platform では、すべてのノードが、プラットフォームコンテナのイメージをプルし、Telemetry データを Red Hat に提供するためにインターネットへの直接のアクセスが必要です。

ロードバランサー

OpenShift Container Platform をインストールする前に、2つの Layer 4 ロードバランサーをプロビジョニングする必要があります。API には1つのロードバランサーが必要で、デフォルトの Ingress コントローラーには、Ingress をアプリケーションに提供する 2 番目のロードバランサーが必要です。

ポート	マシン	内部	外部	説明
6443	ブートストラップおよびコントロールプレーン。ブートストラップマシンがクラスターのコントロールプレーンを初期化した後に、ブートストラップマシンをロードバランサーから削除します。	x	x	Kubernetes API サーバー
22623	ブートストラップおよびコントロールプレーン。ブートストラップマシンがクラスターのコントロールプレーンを初期化した後に、ブートストラップマシンをロードバランサーから削除します。	x		マシン設定サーバー
443	デフォルトで Ingress ルーター Pod、コンピュート、またはワーカーを実行するマシン。	x	x	HTTPS トラフィック

ポート	マシン	内部	外部	説明
80	デフォルトで Ingress ルーター Pod、コンピュータ、またはワーカーを実行するマシン。	x	x	HTTP トラフィック



注記

Ingress ルーターの作業用の設定が OpenShift Container Platform クラスターに必要です。コントロールプレーンの初期化後に Ingress ルーターを設定する必要があります。

1.1.3.2. ユーザーによってプロビジョニングされるインフラストラクチャーの DNS 要件

以下の DNS レコードは、ユーザーによってプロビジョニングされるインフラストラクチャーを使用する OpenShift Container Platform クラスターに必要です。各レコードで、**<cluster_name>** はクラスター名で、**<base_domain>** は、**install-config.yaml** ファイルに指定するクラスターのベースドメインです。

表1.2 必要な DNS レコード

Component	レコード	説明
Kubernetes API	api.<cluster_name>.<base_domain>	この DNS レコードは、コントロールプレーンマシンのロードバランサーを参照する必要があります。このレコードは、クラスター外のクライアントおよびクラスター内のすべてのノードで解決できる必要があります。

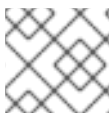
Component	レコード	説明
	api-int.<cluster_name>.<base_domain>	<p>この DNS レコードは、コントロールプレーンマシンのロードバランサーを参照する必要があります。このレコードは、クラスター内のすべてのノードで解決できる必要があります。</p> <div data-bbox="1038 517 1145 1077" style="background-color: black; color: white; padding: 5px; display: inline-block; vertical-align: top;"> <p>重要</p> <p>API サーバーは、Kubernetes に記録されるホスト名でワーカーノードを解決できる必要があります。これがノード名を解決できない場合、プロキシされる API 呼び出しが失敗し、Pod からログを取得できなくなる可能性があります。</p> </div>
Routes	*.apps.<cluster_name>.<base_domain>	<p>Ingress ルーター Pod を実行するマシンをターゲットにするロードバランサーを参照するワイルドカード DNS レコードです。このレコードは、クラスター外のクライアントおよびクラスター内のすべてのノードで解決できる必要があります。</p>

Component	レコード	説明
etcd	etcd-<index>.<cluster_name>.<base_domain>	<p>OpenShift Container Platform では、各 etcd インスタンスの DNS レコードがインスタンスをホストするコントロールプレーンマシンを参照する必要があります。etcd インスタンスは <index> 値によって差別化されます。この値は 0 で始まり、n-1 で終了します。ここで、n はクラスターのコントロールプレーンマシンの数です。DNS レコードはコントロールプレーンマシンのユニキャスト IPv4 アドレスに解決し、レコードはクラスター内のすべてのノードで解決可能である必要があります。</p>
	_etcd-server-ssl._tcp.<cluster_name>.<base_domain>	<p>それぞれのコントロールプレーンマシンについて、OpenShift Container Platform では、そのマシンに優先度 0、重み 10 およびポート 2380 の etcd サーバーの SRV DNS レコードも必要になります。3つのコントロールプレーンマシンを使用するクラスターには以下のレコードが必要です。</p> <pre data-bbox="1042 1227 1436 1989"> # _service._proto.name. TTL class SRV priority weight port target. _etcd-server-ssl._tcp. <cluster_name>. <base_domain> 86400 IN SRV 0 10 2380 etcd- 0.<cluster_name>. <base_domain>. _etcd-server-ssl._tcp. <cluster_name>. <base_domain> 86400 IN SRV 0 10 2380 etcd- 1.<cluster_name>. <base_domain>. _etcd-server-ssl._tcp. <cluster_name>. <base_domain> 86400 IN SRV 0 10 2380 etcd- 2.<cluster_name>. <base_domain>. </pre>

```
# _service._proto.name.          TTL  class SRV priority weight port target.
_etcd-server-ssl._tcp.<cluster_name>.<base_domain> 86400 IN  SRV 0    10   2380 etcd-0.
<cluster_name>.<base_domain>.
_etcd-server-ssl._tcp.<cluster_name>.<base_domain> 86400 IN  SRV 0    10   2380 etcd-1.
<cluster_name>.<base_domain>.
_etcd-server-ssl._tcp.<cluster_name>.<base_domain> 86400 IN  SRV 0    10   2380 etcd-2.
<cluster_name>.<base_domain>.
```

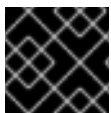
1.1.4. SSH プライベートキーの生成およびエージェントへの追加

クラスターでインストールのデバッグまたは障害復旧を実行する必要がある場合、**ssh-agent** とインストールプログラム の両方に SSH キーを指定する必要があります。



注記

実稼働環境では、障害復旧およびデバッグが必要です。



重要

障害復旧およびデバッグが必要な実稼働環境では、この手順を省略しないでください。

このキーを使用して、ユーザー **core** としてマスターノードに対して SSH を実行できます。クラスターをデプロイする際に、キーは **core** ユーザーの `~/.ssh/authorized_keys` 一覧に追加されます。



注記

[AWS キーペア](#) などのプラットフォームに固有の方法で設定したキーではなく、ローカルキーを使用する必要があります。

手順

1. パスワードなしの認証に設定されている SSH キーがコンピューター上にない場合は、これを作成します。たとえば、Linux オペレーティングシステムを使用するコンピューターで以下のコマンドを実行します。

```
$ ssh-keygen -t rsa -b 4096 -N "" \
-f <path>/<file_name> ①
```

- ① `~/.ssh/id_rsa` などの、SSH キーのパスおよびファイル名を指定します。

このコマンドを実行すると、指定した場所にパスワードを必要としない SSH キーが生成されます。

2. **ssh-agent** プロセスをバックグラウンドタスクとして開始します。

```
$ eval "$(ssh-agent -s)"
Agent pid 31874
```

3. SSH プライベートキーを **ssh-agent** に追加します。


```
$ ssh-add <path>/<file_name> ①
```

```
Identity added: /home/<you>/<path>/<file_name> (<computer_name>)
```

- ① `~/.ssh/id_rsa` などの、SSH プライベートキーのパスおよびファイル名を指定します。

次のステップ

- OpenShift Container Platform をインストールする際に、SSH パブリックキーをインストールプログラムに指定します。

1.1.5. インストールプログラムの取得

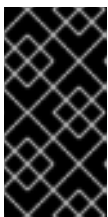
OpenShift Container Platform をインストールする前に、インストールファイルをプロビジョニングマシンにダウンロードします。

前提条件

- Linux を実行するマシンからクラスターをインストールする必要があります (例: Red Hat Enterprise Linux 8)。
- インストールプログラムをダウンロードするには、500 MB のローカルディスク領域が必要です。

手順

1. Red Hat OpenShift Cluster Manager サイトの「[Infrastructure Provider](#)」ページにアクセスします。Red Hat アカウントがある場合は、認証情報を使ってログインします。アカウントがない場合はこれを作成します。
2. 選択するインストールタイプのページに移動し、オペレーティングシステムのインストールプログラムをダウンロードし、ファイルをインストール設定ファイルを保存するディレクトリーに配置します。



重要

インストールプログラムは、クラスターのインストールに使用するコンピューターにいくつかのファイルを作成します。クラスターインストールの完了後は、インストールプログラムおよびインストールプログラムが作成するファイルの両方を保持する必要があります。

3. インストールプログラムを展開します。たとえば、Linux オペレーティングシステムを使用するコンピューターで以下のコマンドを実行します。

```
$ tar xvf <installation_program>.tar.gz
```

4. Red Hat OpenShift Cluster Manager サイトの「[Pull Secret](#)」ページから、インストールプルシークレットを `.txt` ファイルとしてダウンロードします。このプルシークレットを使用し、OpenShift Container Platform コンポーネントのコンテナイメージを提供する、Quay.io などの組み込まれた各種の認証局によって提供されるサービスで認証できます。

1.1.6. CLI のインストール

コマンドラインインターフェースを使用して OpenShift Container Platform と対話するために CLI をインストールすることができます。



重要

以前のバージョンの **oc** をインストールしている場合、これを使用して OpenShift Container Platform 4.3 のすべてのコマンドを実行することはできません。新規バージョンの **oc** をダウンロードし、インストールします。

手順

1. Red Hat OpenShift Cluster Manager サイトの [Infrastructure Provider](#) ページから、選択するインストールタイプのページに移動し、**Download Command-line Tools** をクリックします。
2. オペレーティングシステムおよびアーキテクチャーのフォルダーをクリックしてから、圧縮されたファイルをクリックします。



注記

oc は Linux、Windows、または macOS にインストールできます。

3. ファイルをファイルシステムに保存します。
4. 圧縮ファイルを展開します。
5. これを **PATH** にあるディレクトリーに配置します。

CLI のインストール後は、**oc** コマンドを使用して利用できます。

```
$ oc <command>
```

1.1.7. インストール設定ファイルの手動作成

ユーザーによってプロビジョニングされるインフラストラクチャーを使用する OpenShift Container Platform のインストールでは、インストール設定ファイルを手動で生成する必要があります。

前提条件

- OpenShift Container Platform インストーラープログラムおよびクラスターのアクセストークンを取得します。

手順

1. 必要なインストールアセットを保存するためのインストールディレクトリーを作成します。

```
$ mkdir <installation_directory>
```



重要

ディレクトリを作成する必要があります。ブートストラップ X.509 証明書などの一部のインストールアセットの有効期限は短く設定されているため、インストールディレクトリを再利用することができません。別のクラスターインストールの個別のファイルを再利用する必要がある場合は、それらをインストールディレクトリにコピーすることができます。ただし、インストールアセットのファイル名はリリース間で変更される可能性があります。インストールファイルを以前のバージョンの OpenShift Container Platform からコピーする場合は注意してコピーを行ってください。

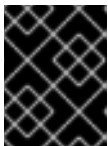
- 以下の **install-config.yaml** ファイルテンプレートをカスタマイズし、これを **<installation_directory>** に保存します。



注記

この設定ファイル **install-config.yaml** に名前を付ける必要があります。

- install-config.yaml** ファイルをバックアップし、これを複数のクラスターをインストールするために使用できるようにします。



重要

install-config.yaml ファイルは、インストールプロセスの次の手順で使用されます。この時点でこれをバックアップする必要があります。

1.1.7.1. ベアメタルのサンプル install-config.yaml ファイル

install-config.yaml ファイルをカスタマイズして、OpenShift Container Platform クラスターのプラットフォームについての詳細を指定するか、または必要なパラメーターの値を変更することができます。

```

apiVersion: v1
baseDomain: example.com 1
compute:
- hyperthreading: Enabled 2 3
  name: worker
  replicas: 0 4
controlPlane:
  hyperthreading: Enabled 5 6
  name: master 7
  replicas: 3 8
metadata:
  name: test 9
networking:
  clusterNetwork:
  - cidr: 10.128.0.0/14 10
    hostPrefix: 23 11
  networkType: OpenShiftSDN
  serviceNetwork: 12
  - 172.30.0.0/16
platform:
  none: {} 13

```

```
fips: false 14
pullSecret: '{"auths": ...}' 15
sshKey: 'ssh-ed25519 AAAA...' 16
```

- 1** クラスターのベースドメイン。すべての DNS レコードはこのベースのサブドメインである必要があります。クラスター名が含まれる必要があります。
- 2 5** **controlPlane** セクションは単一マッピングですが、コンピュートセクションはマッピングのシーケンスになります。複数の異なるデータ構造の要件を満たすには、**compute** セクションの最初の行はハイフン - で始め、**controlPlane** セクションの最初の行はハイフンで始めることができません。どちらのセクションも、現時点では単一のマシンプールを定義しますが、OpenShift Container Platform の今後のバージョンでは、インストール時の複数のコンピュートプールの定義をサポートする可能性があります。1つのコントロールプレーンプールのみが使用されます。
- 3 6 7** 同時マルチスレッドまたは **hyperthreading** を有効/無効にするかどうか。デフォルトでは、同時スレッドはマシンのコアのパフォーマンスを上げるために有効にされます。パラメーター値を **Disabled** に設定するとこれを無効にすることができます。一部のクラスターマシンで同時マルチスレッドを無効にする場合は、これをすべてのクラスターマシンで無効にする必要があります。



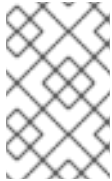
重要

同時スレッドを無効にする場合は、容量計画においてマシンパフォーマンスの大幅な低下が考慮に入れられていることを確認します。

- 4** **replicas** パラメーターの値を **0** に設定する必要があります。このパラメーターはクラスターが作成し、管理するワーカーの数を制御します。これは、ユーザーによってプロビジョニングされるインフラストラクチャーを使用する場合にクラスターが実行しない機能です。OpenShift Container Platform のインストールが終了する前に、クラスターが使用するワーカーマシンを手動でデプロイする必要があります。
- 8** クラスターに追加するコントロールプレーンマシンの数。クラスターをこの値をクラスターの etcd エンドポイント数として使用するため、値はデプロイするコントロールプレーンマシンの数に一致する必要があります。
- 9** DNS レコードに指定したクラスター名。
- 10** Pod IP アドレスの割り当てに使用する IP アドレスのブロック。このブロックは既存の物理ネットワークと重複できません。これらの IP アドレスは Pod ネットワークに使用され、外部ネットワークから Pod にアクセスする必要がある場合には、ロードバランサーおよびルーターを、トラフィックを管理するように設定します。
- 11** それぞれの個別ノードに割り当てるサブネットプレフィックスの長さ。たとえば、**hostPrefix** が **23** に設定され、各ノードに指定の **cidr** から **/23** サブネットが割り当てられます (510 (2³²⁻²³ - 2) Pod IP アドレスが許可されます)。外部ネットワークからのノードへのアクセスを提供する必要がある場合には、ロードバランサーおよびルーターを、トラフィックを管理するように設定します。
- 12** サービス IP アドレスに使用する IP アドレスプール。1つの IP アドレスプールのみを入力できます。外部ネットワークからサービスにアクセスする必要がある場合、ロードバランサーおよびルーターを、トラフィックを管理するように設定します。
- 13** プラットフォームを **none** に設定する必要があります。ベアメタルインフラストラクチャー用に追加のプラットフォーム設定変数を指定することはできません。
- 14** FIPS モードを有効または無効にするかどうか。デフォルトでは、FIPS モードは有効にされません。FIPS モードが有効にされている場合、OpenShift Container Platform が実行される Red Hat

Enterprise Linux CoreOS (RHCOS) マシンがデフォルトの Kubernetes 暗号スイートをバイパスし、代わりに RHCOS で提供される FIPS 検証済み暗号モジュールを使用します。

- 15 Red Hat OpenShift Cluster Manager サイトの「[Pull Secret](#)」ページから取得したプルシークレット。このプルシークレットを使用し、OpenShift Container Platform コンポーネントのコンテナイメージを提供する、Quay.io などの組み込まれた各種の認証局によって提供されるサービスで認証できます。
- 16 Red Hat Enterprise Linux CoreOS (RHCOS) の **core** ユーザーのデフォルト SSH キーの公開部分。



注記

インストールのデバッグまたは障害復旧を実行する必要がある実稼働用の OpenShift Container Platform クラスターでは、**ssh-agent** プロセスが使用する SSH キーを指定します。

1.1.8. Kubernetes マニフェストおよび Ignition 設定ファイルの作成

一部のクラスター定義ファイルを変更し、クラスターマシンを手動で起動する必要があるため、クラスターがマシンを作成するために必要な Kubernetes マニフェストと Ignition 設定ファイルを生成する必要があります。



重要

インストールプログラムが生成する Ignition 設定ファイルには、24 時間が経過すると期限切れになる証明書が含まれます。最初の証明書のローテーションが正常に実行されるようにするには、クラスターのインストールを完了し、クラスターを動作が低下していない状態で 24 時間実行し続ける必要があります。

前提条件

- OpenShift Container Platform インストールプログラムを取得します。
- **install-config.yaml** インストール設定ファイルを作成します。

手順

1. クラスターの Kubernetes マニフェストを生成します。

```
$ ./openshift-install create manifests --dir=<installation_directory> 1
```

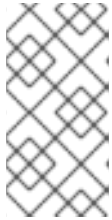
```
WARNING There are no compute nodes specified. The cluster will not fully initialize without compute nodes.
```

```
INFO Consuming "Install Config" from target directory
```

- 1 **<installation_directory>** については、作成した **install-config.yaml** ファイルが含まれるインストールディレクトリーを指定します。

インストールプロセスの後の部分で独自のコンピュータマシンを作成するため、この警告を無視しても問題がありません。

2. **manifests/cluster-scheduler-02-config.yml** Kubernetes マニフェストファイルを変更し、Pod がコントロールプレーンマシンにスケジュールされないようにします。
 - a. **manifests/cluster-scheduler-02-config.yml** ファイルを開きます。
 - b. **mastersSchedulable** パラメーターを見つけ、その値を **False** に設定します。
 - c. ファイルを保存し、終了します。



注記

現時点では、[Kubernetes の制限](#)により、コントロールプレーンマシンで実行されるルーター Pod に Ingress ロードバランサーがアクセスすることができません。この手順は、OpenShift Container Platform の今後のマイナーバージョンで不要になる可能性があります。

3. Ignition 設定ファイルを取得します。

```
$ ./openshift-install create ignition-configs --dir=<installation_directory> ①
```

- ① **<installation_directory>** については、同じインストールディレクトリーを指定します。

以下のファイルはディレクトリーに生成されます。

```
.
├── auth
│   ├── kubeadmin-password
│   └── kubeconfig
├── bootstrap.ign
├── master.ign
├── metadata.json
└── worker.ign
```

1.1.9. Red Hat Enterprise Linux CoreOS (RHCOS) マシンの作成

プロビジョニングする IBM Z インフラストラクチャーにクラスターをインストールする前に、クラスターが使用する RHCOS を z/VM ゲスト仮想マシンにインストールする必要があります。マシンを作成するには、以下の手順を実行します。

前提条件

- 作成するマシンがアクセスできるプロビジョニングマシンで稼働している FTP サーバー。

手順

1. プロビジョニングマシンで Linux にログインします。
2. [RHCOS イメージミラー](#)から Red Hat Enterprise Linux CoreOS インストールファイルをダウンロードします。



重要

RHCOS イメージは OpenShift Container Platform の各リリースごとに変更されない可能性があります。インストールする OpenShift Container Platform バージョンと等しいか、それ以下のバージョンの中で最も新しいバージョンのイメージをダウンロードする必要があります。利用可能な場合は、OpenShift Container Platform バージョンに一致するイメージのバージョンを使用します。

以下のファイルをダウンロードします。

- initramfs: **rhcos-<version>-installer-initramfs.img**
 - kernel: **rhcos-<version>-installer-kernel**
 - RHCOS をインストールするディスクのオペレーティングシステムイメージ。このタイプは仮想マシンによって異なる場合があります。
rhcos-<version>-s390x-metal-dasd.raw.gz (DASD 用)
rhcos-<version>-s390x-metal-zfcp.raw.gz (FCP 用)
3. パラメーターファイルを作成します。以下のパラメーターは特定の仮想マシンに固有のものです。
- **coreos.inst.install_dev=** の場合、DASD インストールに **dasda** を指定するか、または FCP に **sda** を指定します。FCP には **zfcp.allow_lun_scan=0** が必要なことに注意してください。
 - **rd.dasd=** の場合、RHCOS がインストールされる DASD を指定します。
 - **rd.zfcp=<adapter>,<wwpn>,<lun>** は、RHCOS をインストールする FCP ディスクを指定します。
 - **ip=**には、以下の7つのエントリーを指定します。
 - i. マシンの IP アドレス。
 - ii. 空の文字列。
 - iii. ゲートウェイ。
 - iv. ネットマスク。
 - v. **hostname.domainname** 形式のマシンホストおよびドメイン名。RHCOS に決定し、設定させる場合は、この値を省略します。
 - vi. ネットワークインターフェース名。RHCOS に決定し、設定させる場合は、この値を省略します。
 - vii. 静的 IP アドレスを使用する場合、空の文字列になります。
 - **coreos.inst.ignition_url=** の場合、マシンロールの Ignition ファイルを指定します。ブートストラップマシンの Ignition ファイルは **bootstrap-0** と呼ばれ、マスター Ignition ファイルは 2 から 0 までの番号が付けられ、ワーカーの Ignition ファイルに 0 以降の番号が付けられます。その他のパラメーターはそのまま利用できます。
ブートストラップマシンのパラメーターファイルのサンプル **bootstrap-0.parm**:

```
rd.neednet=1 coreos.inst=yes coreos.inst.install_dev=dasda coreos.inst.image_url=ftp://
```

```
cl1.provide.example.com:8080/assets/rhcos-42.80.20191105.0-metal-dasd.raw.gz
coreos.inst.ignition_url=ftp://cl1.provide.example.com:8080/ignition-bootstrap-0
ip=172.18.78.2::172.18.78.1:255.255.255.0:::none nameserver=172.18.78.1
rd.znet=qeth,0.0.bdf0,0.0.bdf1,0.0.bdf2,layer2=1,portno=0 zfcplib.allow_lun_scan=0
cio_ignore=all,
lcondev rd.dasd=0.0.3490
```

- FTP などを使用し、initramfs、kernel、パラメーターファイル、および RHCOS イメージを z/VM に転送します。FTP でファイルを転送し、仮想リーダーから起動する方法については、「[z/VM 環境へのインストール](#)」を参照してください。
- ブートストラップノードになる z/VM ゲスト仮想マシンの仮想リーダーに対してファイルの punch を実行します。
IBM Knowledge Center で [PUNCH](#) を参照してください。

ヒント

CP PUNCH コマンドを使用するか、Linux を使用している場合は、`vmur` コマンドを使用して 2 つの z/VM ゲスト仮想マシン間でファイルを転送できます。

- ブートストラップマシンで CMS にログインします。
- リーダーからブートストラップマシンに対して IPL を実行します。

```
$ ipl c
```

IBM Knowledge Center で [IPL](#) を参照してください。

- クラスタ内の他のマシンについてこの手順を繰り返します。

1.1.10. クラスタの作成

OpenShift Container Platform クラスタを作成するには、ブートストラッププロセスが、インストールプログラムで生成した Ignition 設定ファイルを使用してプロビジョニングしたマシンで完了するのを待機します。

前提条件

- クラスタに必要なインフラストラクチャーを作成すること。
- インストールプログラムを取得し、クラスタの Ignition 設定ファイルを生成していること。
- クラスタの RHCOS マシンを作成するために Ignition 設定ファイルを使用していること。
- 使用するマシンでインターネットに直接アクセスできること。

手順

- ブートストラッププロセスをモニターします。

```
$ ./openshift-install --dir=<installation_directory> wait-for bootstrap-complete \ ❶
--log-level=info ❷
```

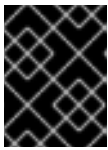


```
INFO Waiting up to 30m0s for the Kubernetes API at https://api.test.example.com:6443...
INFO API v1.14.6+c4799753c up
INFO Waiting up to 30m0s for the bootstrap-complete event...
```

- 1 **<installation_directory>** については、インストールファイルを保存したディレクトリへのパスを指定します。
- 2 異なるインストールの詳細情報を表示するには、**info** ではなく、**warn**、**debug**、または **error** を指定します。

Kubernetes API サーバーでこれがコントロールプレーンマシンにブートストラップされていることを示すシグナルが出されるとコマンドは成功します。

2. ブートストラッププロセスが完了したら、ブートストラップマシンをロードバランサーから削除します。



重要

この時点で、ブートストラップマシンをロードバランサーから削除する必要があります。さらに、マシン自体を削除し、再フォーマットすることができます。

1.1.11. クラスターへのログイン

クラスター **kubeconfig** ファイルをエクスポートし、デフォルトシステムユーザーとしてクラスターにログインできます。**kubeconfig** ファイルには、クライアントを正しいクラスターおよび API サーバーに接続するために CLI で使用されるクラスターについての情報が含まれます。このファイルはクラスターに固有のファイルであり、OpenShift Container Platform のインストール時に作成されます。

前提条件

- OpenShift Container Platform クラスターのデプロイ。
- **oc** CLI のインストール。

手順

1. **kubeadmin** 認証情報をエクスポートします。

```
$ export KUBECONFIG=<installation_directory>/auth/kubeconfig 1
```

- 1 **<installation_directory>** については、インストールファイルを保存したディレクトリへのパスを指定します。

2. エクスポートされた設定を使用して、**oc** コマンドを正常に実行できることを確認します。

```
$ oc whoami
system:admin
```

1.1.12. マシンの CSR の承認

マシンをクラスターに追加する際に、追加したそれぞれのマシンについて2つの保留状態の証明書署名要求 (CSR) が生成されます。これらの CSR が承認されていることを確認するか、または必要な場合はそれらを承認してください。

前提条件

- マシンをクラスターに追加していること。
- `jq` パッケージのインストール。

手順

1. クラスターがマシンを認識していることを確認します。

```
# oc get nodes

NAME                STATUS ROLES  AGE  VERSION
master-0.cl1mstr0.example.com Ready  master  20h  v1.14.6+888f9c630
master-1.cl1mstr1.example.com Ready  master  20h  v1.14.6+888f9c630
master-2.cl1mstr2.example.com Ready  master  20h  v1.14.6+888f9c630
worker-0.cl1wrk00.example.com Ready  worker  20h  v1.14.6+888f9c630
worker-1.cl1wrk01.example.com Ready  worker  20h  v1.14.6+888f9c630
```

出力には作成したすべてのマシンが一覧表示されます。

2. 保留中の証明書署名要求 (CSR) を確認し、クラスターに追加したそれぞれのマシンのクライアントおよびサーバー要求に **Pending** または **Approved** ステータスが表示されていることを確認します。

```
$ oc get csr

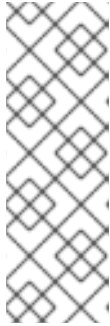
NAME          AGE  REQUESTOR                                     CONDITION
csr-8b2br    15m  system:serviceaccount:openshift-machine-config-operator:node-bootstraptrapper Pending 1
csr-8vnps    15m  system:serviceaccount:openshift-machine-config-operator:node-bootstraptrapper Pending
csr-bfd72    5m26s system:node:ip-10-0-50-126.us-east-2.compute.internal Pending 2
csr-c57lv    5m26s system:node:ip-10-0-95-157.us-east-2.compute.internal Pending
...
```

1 クライアント要求の CSR。

2 サーバー要求の CSR。

この例では、2つのマシンがクラスターに参加しています。この一覧にはさらに多くの承認された CSR が表示される可能性があります。

3. 追加したマシンの保留中の CSR すべてが **Pending** ステータスになった後に CSR が承認されない場合には、クラスターマシンの CSR を承認します。



注記

CSR のローテーションは自動的に実行されるため、クラスターにマシンを追加後1時間以内に CSR を承認してください。1時間以内に承認しない場合には、証明書のローテーションが行われ、各ノードに3つ以上の証明書が存在するようになります。これらの証明書すべてを承認する必要があります。最初の CSR の承認後、後続のノードクライアント CSR はクラスターの **kube-controller-manger** によって自動的に承認されます。kubelet 提供証明書の要求を自動的に承認する方法を実装する必要があります。

- それらを個別に承認するには、それぞれの有効な CSR について以下のコマンドを実行します。

```
$ oc adm certificate approve <csr_name> ❶
```

- ❶ **<csr_name>** は、現行の CSR の一覧からの CSR の名前です。

- すべての CSR が有効な場合、以下のコマンドを実行してそれらすべてを承認します。

```
$ oc get csr -ojson | jq -r '.items[] | select(.status == {} ) | .metadata.name' | xargs oc adm certificate approve
```

1.1.13. Operator の初期設定

コントロールプレーンの初期化後に、一部の Operator を利用可能にするためにそれらをすぐに設定する必要があります。

前提条件

- コントロールプレーンが初期化されていること。

手順

1. クラスターコンポーネントがオンラインになることを確認します。

```
$ watch -n5 oc get clusteroperators
```

NAME	VERSION	AVAILABLE	PROGRESSING	DEGRADED	SINCE
authentication	4.3.0	True	False	False	69s
cloud-credential	4.3.0	True	False	False	12m
cluster-autoscaler	4.3.0	True	False	False	11m
console	4.3.0	True	False	False	46s
dns	4.3.0	True	False	False	11m
image-registry	4.3.0	True	False	False	5m26s
ingress	4.3.0	True	False	False	5m36s
kube-apiserver	4.3.0	True	False	False	8m53s
kube-controller-manager	4.3.0	True	False	False	7m24s
kube-scheduler	4.3.0	True	False	False	12m
machine-api	4.3.0	True	False	False	12m
machine-config	4.3.0	True	False	False	7m36s
marketplace	4.3.0	True	False	False	7m54m
monitoring	4.3.0	True	False	False	7h54s

network	4.3.0	True	False	False	5m9s
node-tuning	4.3.0	True	False	False	11m
openshift-apiserver	4.3.0	True	False	False	11m
openshift-controller-manager	4.3.0	True	False	False	5m943s
openshift-samples	4.3.0	True	False	False	3m55s
operator-lifecycle-manager	4.3.0	True	False	False	11m
operator-lifecycle-manager-catalog	4.3.0	True	False	False	11m
service-ca	4.3.0	True	False	False	11m
service-catalog-apiserver	4.3.0	True	False	False	5m26s
service-catalog-controller-manager	4.3.0	True	False	False	5m25s
storage	4.3.0	True	False	False	5m30s

2. 利用不可の Operator を設定します。

1.1.13.1. イメージレジストリーストレージの設定

image-registry Operator が利用できない場合、そのストレージを設定する必要があります。実稼働クラスターに必要な PersistentVolume の設定方法と、実稼働用ではないクラスターにのみ使用できる空のディレクトリーをストレージの場所として設定する方法が表示されます。

1.1.13.1.1. ベアメタルの場合のレジストリーストレージの設定

クラスター管理者は、インストール後にレジストリーをストレージを使用できるように設定する必要があります。

前提条件

- クラスター管理者のパーミッション。
- ベアメタル上のクラスター。
- クラスターの永続ストレージをプロビジョニングします。プライベートイメージレジストリーをデプロイするには、ストレージで ReadWriteMany アクセスモードを指定する必要があります。
- 容量は「100Gi」以上である。

手順

1. レジストリーをストレージを使用できるように設定するには、**configs.imageregistry/cluster** リソースの **spec.storage.pvc** を変更します。
2. レジストリー Pod がないことを確認します。

```
$ oc get pod -n openshift-image-registry
```



注記

ストレージタイプが **emptyDIR** の場合、レプリカ数が **1** を超えることはありません。ストレージタイプが **NFS** で、レジストリー Pod を **replica>1** を設定してスケールアップする必要がある場合、**no_wdelay** マウントオプションを有効にする必要があります。以下は例になります。

```
# cat /etc/exports
/mnt/data *(rw,sync,no_wdelay,no_root_squash,insecure,fsid=0)
sh-4.3# exportfs -rv
exporting */mnt/data
```

3. レジストリー設定を確認します。

```
$ oc edit configs.imageregistry.operator.openshift.io
```

```
storage:
pvc:
claim:
```

claim フィールドを空のままにし、**image-registry-storage** PVC の自動作成を可能にします。

4. **clusteroperator** ステータスを確認します。

```
$ oc get clusteroperator image-registry
```

1.1.13.1.2. 実稼働以外のクラスターでのイメージレジストリーのストレージの設定

イメージレジストリー Operator のストレージを設定する必要があります。実稼働用以外のクラスターの場合、イメージレジストリーは空のディレクトリーに設定することができます。これを実行する場合、レジストリーを再起動するとすべてのイメージが失われます。

手順

- イメージレジストリーストレージを空のディレクトリーに設定するには、以下を実行します。

```
$ oc patch configs.imageregistry.operator.openshift.io cluster --type merge --patch '{"spec": {"storage":{"emptyDir":{}}}'
```



警告

実稼働用以外のクラスターにのみこのオプションを設定します。

イメージレジストリー Operator がそのコンポーネントを初期化する前にこのコマンドを実行する場合、**oc patch** コマンドは以下のエラーを出して失敗します。

```
Error from server (NotFound): configs.imageregistry.operator.openshift.io "cluster" not found
```

数分待機した後に、このコマンドを再び実行します。

1.1.14. ユーザーによってプロビジョニングされるインフラストラクチャーでのインストールの完了

Operator 設定の完了後に、提供するインフラストラクチャーでのクラスターのインストールを終了できます。

前提条件

- コントロールプレーンが初期化されていること。
- Operator の初期設定を完了していること。

手順

1. すべてのクラスターコンポーネントがオンラインであることを確認します。

```
$ watch -n5 oc get clusteroperators
```

NAME	VERSION	AVAILABLE	PROGRESSING	DEGRADED	SINCE
authentication	4.3.0	True	False	False	10m
cloud-credential	4.3.0	True	False	False	22m
cluster-autoscaler	4.3.0	True	False	False	21m
console	4.3.0	True	False	False	10m
dns	4.3.0	True	False	False	21m
image-registry	4.3.0	True	False	False	16m
ingress	4.3.0	True	False	False	16m
kube-apiserver	4.3.0	True	False	False	19m
kube-controller-manager	4.3.0	True	False	False	18m
kube-scheduler	4.3.0	True	False	False	22m
machine-api	4.3.0	True	False	False	22m
machine-config	4.3.0	True	False	False	18m
marketplace	4.3.0	True	False	False	18m
monitoring	4.3.0	True	False	False	18m
network	4.3.0	True	False	False	16m
node-tuning	4.3.0	True	False	False	21m
openshift-apiserver	4.3.0	True	False	False	21m
openshift-controller-manager	4.3.0	True	False	False	17m
openshift-samples	4.3.0	True	False	False	14m
operator-lifecycle-manager	4.3.0	True	False	False	21m
operator-lifecycle-manager-catalog	4.3.0	True	False	False	21m
service-ca	4.3.0	True	False	False	21m
service-catalog-apiserver	4.3.0	True	False	False	16m
service-catalog-controller-manager	4.3.0	True	False	False	16m
storage	4.3.0	True	False	False	16m

すべてのクラスター Operator が **AVAILABLE** の場合、インストールを完了することができます。

2. クラスターの完了をモニターします。

```
$ ./openshift-install --dir=<installation_directory> wait-for install-complete 1
INFO Waiting up to 30m0s for the cluster to initialize...
```

- 1 **<installation_directory>** については、インストールファイルを保存したディレクトリへのパスを指定します。

Cluster Version Operator が Kubernetes API サーバーから OpenShift Container Platform クラスターのデプロイを終了するとコマンドは成功します。



重要

インストールプログラムが生成する Ignition 設定ファイルには、24 時間が経過すると期限切れになる証明書が含まれます。最初の証明書のローテーションが正常に実行されるようにするには、クラスターを動作が低下していない状態で 24 時間実行し続ける必要があります。

3. Kubernetes API サーバーが Pod と通信していることを確認します。
- a. すべての Pod の一覧を表示するには、以下のコマンドを使用します。

```
$ oc get pods --all-namespaces
```

NAMESPACE	NAME	READY	STATUS
RESTARTS	AGE		
openshift-apiserver-operator	openshift-apiserver-operator-85cb746d55-zqhs8	1/1	Running
1	9m		
openshift-apiserver	apiserver-67b9g	1/1	Running
3m			
openshift-apiserver	apiserver-ljcmx	1/1	Running
1m			
openshift-apiserver	apiserver-z25h4	1/1	Running
2m			
openshift-authentication-operator	authentication-operator-69d5d8bf84-vh2n8	1/1	Running
0	5m		
...			

- b. 以下のコマンドを使用して、直前のコマンドの出力に一覧表示される Pod のログを表示します。

```
$ oc logs <pod_name> -n <namespace> 1
```

- 1 直前のコマンドの出力にあるように、Pod 名および namespace を指定します。

Pod のログが表示される場合、Kubernetes API サーバーはクラスターマシンと通信できません。

次のステップ

- [クラスターをカスタマイズ](#) します。
- 必要な場合は、[リモートの健全性レポートをオプトアウト](#) することができます。

